BELLCOMM, INC.
1100 Seventeenth Street, N.W. Washington, D.C. 20036

SUBJECT: Triplicated Interface Element for Triply Redundant Modules in Digital Devices - Case 730

DATE: February 5, 1968

FROM: D. O. Baechler

ABSTRACT

This memorandum describes two Triplicated Interface Elements (TIE's), either of which could be used to provide an interface between two sets of triply redundant modules in a digital device. Each of the TIE's can be operated in a mode that gives a majority vote on the triplicated outputs, and each has several other modes of operation in which the outputs are chosen under control of external, software-operated control signals. Indications of disagreements among triplicated outputs are also given.

Each element is described in the form of a minimized logic block diagram. These configurations can be used to determine the feasibility of incorporating such elements in a triply modular redundant computer.

BELLCOMM, INC.
1100 Seventeenth Street, N.W. Washington, D.C. 20036

SUBJECT: Triplicated Interface Element for Triply Redundant Modules in Digital Devices - Case 730

DATE: February 5, 1968

FROM: D. O. Baechler

MEMORANDUM FOR FILE

I. INTRODUCTION

Redundancy can be used to increase the reliability of digital electronic equipment. A method of accomplishing this is to divide the equipment into modules, triplicate the modules, and use a voting circuit to determine the majority output of each triplicated group. The majority is then taken to be the "correct" output.

The Saturn Launch Vehicle Digital Computer uses a similar type of redundancy except that the voters are triplicated, and error (i.e., disagreement) detection and limited fault location are provided. It has been suggested# that this type of technique is applicable to general digital circuits, and that voters of this type might also incorporate control signals that would cause the circuit to operate in a simplex mode, in which each of the triplicated modules operates independently of the other two.

This memorandum describes a generalized Interface Element (IE) and a simplified Interface Element, both of which have a majority voting mode, a simplex mode, and other modes that are described later in this memorandum. Both elements also provide for error detection and fault location. The difference between the two IE's is the number of "other" modes in each.

Figure 1 is a block diagram of the type of circuit used in the Saturn vehicle. M_1, M_2, and M_3 are identical digital devices. Their respective binary outputs, A_1, A_2, and A_3, are voted on by voters V_1, V_2, and V_3 to yield the binary outputs A_1^*, A_2^*, and A_3^*. These outputs, A_1^*, A_2^*, and A_3^*, are equal to one another and have the same logic value as the majority of the binary signals A_1, A_2, and A_3. Either of the IE's described in this report could be used in place of the

---

#"Separable Redundant Computers," L. D. Amdahl and L. T. Mast, The Rand Corporation, Memorandum RM-4573-NASA, September, 1965.

voters in Figure 1. The resulting Triplicated Interface Elements (TIE's) are shown in Figures 2 and 4.

II. DESIGN OF GENERALIZED INTERFACE ELEMENT

A block diagram of a TIE is shown in Figure 2. There are three interface elements in the TIE. In addition to the input signals A_1, A_2, and A_3, the "i"th interface element has as inputs the control signals R_i and S_i, the output A_i^* and indicators X_i, Y_i, and Z_i. The output A_i^* of the IE depends on the control signals R_i and S_i in the following manner:

R_i	S_i	A_i^*
0	0	$A_1A_2+A_1A_3+A_2A_3$ (Majority)
0	1	A_1
1	0	A_2
1	1	A_3

The signals X_i, Y_i, and Z_i are disagreement indicators. X_i = 1 indicates that A_1 disagrees with A_2 and with A_3. Y_i = 1 indicates that A_2 disagrees with A_1 and with A_3. And Z_i= 1 indicates that A_3 disagrees with A_1 and A_2. Furthermore, once disagreement occurs, the indicator remains set until reset by software through control signals R_i and S_i. Therefore, X_i, Y_i, and Z_i are functions of their previous values as well as being functions of A_1, A_2, A_3, R_i, and S_i. The previous values of X_i, Y_i, and Z_i will be referred to as x_i, y_i, and z_i.

The 64 possible combinations of A_1, A_2, A_3, x_i, y_i, and z_i are given in Table 1, and, for each combination, the corresponding values of A_i^*, X_i, Y_i, and Z_i are given for each of the four possible combinations of R_i and S_i. The table can be briefly explained as follows:

If R_i, S_i = 0, 0: Software is commanding the output to be the majority of the inputs. X_i = 1 if A_1 disagrees with A_2 and with A_3, or if x_i = 1 (which would have resulted from a previous disagreement.) Similarly, Y_i = 1 if A_2 disagrees with A_1 and with A_3, or if y_i = 1. And Z_i = 1 if A_3 disagrees with A_1 and with A_2, or if z_i = 1.

R_i, S_i = 0, 1: Software is commanding the output to be equal to A_1 as a result of previously sensing that Z_i = 1 (i.e., A_3 disagreed with A_1 and with A_2). In an effort to reduce software requirements, it was decided to reset Z_i so that Z_i = 0, and to inhibit the set signal so that, even though A_3 may continue to disagree with A_1 and A_2, Z_i will not be set to 1. X_i continues to indicate when A_1 disagrees with both A_2 and A_3, and Y_i continues to indicate when A_2 disagrees with both A_1 and A_3. Therefore, X_i and Y_i will both remain zero if and only if A_1 and A_2 agree.

R_i, S_i = 1, 1: Same as for R_i, S_i = 0, 1 except that software has chosen A_3 for the output as a result of A_2 having previously disagreed with A_1 and A_3 (i.e., Y_i = 1). Y_i is reset, its set signal inhibited, and X_i and Z_i indicate disagreement between A_1 and A_3.

R_i, S_i = 1, 0: Same as for R_i, S_i = 0, 1 except A_2 is the output, A_1 has disagreed (X_i = 1), X_i gets reset, its set signal is inhibited and Y_i and Z_i continue comparing A_2 and A_3.

The resulting logic equations are shown in Table 1. Figure 3 shows an implementation of the generalized IE using NAND gates. Although no timing inputs are shown in the figure, it would be necessary to strobe each of the second-level gates to avoid critical races. This would increase the total number of inputs from 69 to 84.

Figure 3 is the circuit for a single IE. Three identical IE's form a TIE. A TIE has six control signals (R_1, S_1, R_2, S_2, R_3, S_3) as inputs and nine indicator signals (X_1, Y_1, Z_1, X_2, Y_2, Z_2, X_3, Y_3, Z_3) as outputs. Each IE can be independently controlled to have its output, A_i^*, set equal to the majority ($M = A_1A_2+A_2A_3+A_1A_3$) or it can be set equal to A_1, or to A_2 or to A_3. Since the three A_i^* outputs can be, independently, any of the four possible variables, there are a total of $4^3 = 64$ possible combinations of variables at the three outputs. This is a very general interface element, and it is not clear how important it is to have all the possible combinations of outputs available. For instance, consider the combinations listed in the table below:

A_1^*	A_2^*	A_3^*
Majority	Majority	Majority
A_1	A_2	A_3
A_1	A_1	A_2
A_3	A_1	A_3
A_3	A_1	A_1
A_3	A_2	A_1

The first combination, each output indicating the majority of the inputs, is of interest. The next combination is also of interest since it represents operation as though there were three simplex machines. The remainder of the combinations could be of interest in certain checking exercises, but their utility is somewhat vague. It is of particular interest to see what price is paid for these combinations whose usefulness is not established. This can be done by comparing the generalized design with a simpler design that has as outputs a subset from the combinations of outputs that are available in the generalized IE. The design of such a simplified interface element is described below.

III. DESIGN OF SIMPLIFIED INTERFACE ELEMENT

The first step in the design of a simplified IE is the choice of the subset of output combinations. The subset chosen for this design is shown below. It is one of several reasonable, interesting, and small subsets.

QRS	A_1^*	A_2^*	A_3^*
000	Majority	Majority	Majority
001	A_1	A_1	A_1
010	A_2	A_2	A_2
011	A_3	A_3	A_3
100	A_1	A_2	A_3

The control signals Q, R, and S in this case have no subscripts; the same three control signals go to each IE in the TIE, as shown in Figure 4. Although each IE is fabricated exactly alike, the connections from the inputs are different and the outputs are not the same, hence it is not possible to use the notation A_i^* as the output from the "i"th IE. Therefore, the design for the first IE is discussed, and the design for the second IE can be found by substituting the subscripts 2, 3, and 1 for the subscripts 1, 2, and 3, respectively. Similarly, the third IE would have substituted the subscripts 3, 1 and 2.

The values of the indicators in this simplified IE are the same as for the general IE for Q = 0. For Q = 1, they are the same as for the R, S = 0, 0 case in the general IE. But note that the first IE contains only X, the second IE contains only Y and the third IE contains only Z.

The values of the outputs in this simplified IE are the same as for the general IE for Q = 0. For Q = 1, the output of the first element is A_1, that of the second element is A_2 and that of the third is A_3. The resulting logic equations are as follows:

$$A_1^* = \overline{R}(A_1A_2) + \overline{Q}\overline{S}(A_2A_3) + \overline{R}(A_1A_3) + QA_1 + \overline{R}SA_1 + R\overline{S}A_2 + RSA_3$$

$$A_2^* = \overline{R}(A_1A_2) + \overline{S}(A_2A_3) + \overline{Q}\overline{R}(A_1A_3) + QA_2 + \overline{R}SA_1 + R\overline{S}A_2 + RSA_3$$

$$A_3^* = \overline{Q}\overline{R}(A_1A_2) + \overline{S}(A_2A_3) + \overline{R}(A_1A_3) + QA_3 + \overline{R}SA_1 + R\overline{S}A_2 + RSA_3$$

$$X = [Q + \overline{R} + S]\ [x + A_1\overline{A}_2\overline{A}_3 + \overline{A}_1A_2A_3]$$

$$Y = [Q + \overline{R} + \overline{S}]\ [y + A_1\overline{A}_2A_3 + \overline{A}_1A_2\overline{A}_3]$$

$$Z = [Q + R + \overline{S}]\ [z + A_1A_2\overline{A}_3 + \overline{A}_1\overline{A}_2A_3]$$

The minimized logic block diagram for A_1^* and X is shown in Figure 5. As was the case for the generalized Interface Element, the problem of critical races could be overcome by strobing each of the second level gates with an appropriate timing signal. This would increase the total number of inputs from 45 to 55.

IV. COMMENTS

A generalized IE requires 22 NAND gates and a simplified IE requires 13 NAND gates. Since there are three IE's in a TIE, the requirement for a TIE is 66 or 39 NAND gates, depending on the type of TIE used. Generally, each of the triplicated modules, M_1, M_2, and M_3 shown in Figures 2 and 4, would have more than one output, and each output would require a TIE of the complexity mentioned above. The relative complexity of the "voter" to the triplicated module is an important factor in determining how big to make the module before interposing a TIE.

On missions of long duration it may be desirable for a variety of reasons to be able to operate a triply redundant computer in a simplex mode with power off the other two parts. Although no rigorous analysis of this possibility has been made for the circuits described in this report, a cursory examination yields no reasons that would make such a scheme unacceptable.

1031-DOB-sel

D. O. Baechler

BELLCOMM, INC.

Subject: Triplicated Interface Element for Triply Redundant Modules in Digital Devices - Case 730

From: D. O. Baechler

Distribution List

NASA Headquarters

Mr. T. A. Keegan/MA

MSC

Messrs. C. Frasier/EG44
G. Xenakis/EG

MSFC

Mr. F. Wojtalik/R-ASTR-N

ERC

Messrs. D. VanMeter/KC
G. Wang/KCT

Bellcomm

Messrs. A. P. Boysen, Jr.
D. R. Hagner
W. C. Hittinger
B. T. Howard
C. M. Klingman
J. Z. Menard
I. D. Nehama
I. M. Ross
P. S. Schaenman
R. L. Wagner
All Members of Department 1031
Department 1023 ←COPY TO
Central File
Library

TABLE I. VALUES OF A_i^*, X_i, Y_i, AND Z_i

	A_1	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101
	A_2	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011
	A_3	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111
	x	0000	0000	1111	1111	0000	0000	1111	1111	0000	0000	1111	1111	0000	0000	1111	1111
	y	0000	0000	0000	0000	1111	1111	1111	1111	0000	0000	0000	0000	1111	1111	1111	1111
	z	0000	0000	0000	0000	0000	0000	0000	0000	1111	1111	1111	1111	1111	1111	1111	1111
	A_i^*	0001	0111	0001	0111	0001	0111	0001	0111	0001	0111	0001	0111	0001	0111	0001	0111
$R_i = 0$	X_i	0100	0010	1111	1111	0100	0010	1111	1111	0100	0010	1111	1111	0100	0010	1111	1111
$S_i = 0$	Y_i	0010	0100	0010	0100	1111	1111	1111	1111	0010	0100	0010	0100	1111	1111	1111	1111
	Z_i	0001	1000	0001	1000	0001	1000	0001	1000	1111	1111	1111	1111	1111	1111	1111	1111
	A_i^*	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101
$R_i = 0$	X_i	0100	0010	1111	1111	0100	0010	1111	1111	0100	0010	1111	1111	0100	0010	1111	1111
$S_i = 1$	Y_i	0010	0100	0010	0100	1111	1111	1111	1111	0010	0100	0010	0100	1111	1111	1111	1111
	Z_i	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	A_i^*	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111	0000	1111
$R_i = 1$	X_i	0100	0010	1111	1111	0100	0010	1111	1111	0100	0010	1111	1111	0100	0010	1111	1111
$S_i = 1$	Y_i	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	Z_i	0001	1000	0001	1000	0001	1000	0001	1000	1111	1111	1111	1111	1111	1111	1111	1111
	A_i^*	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011
$R_i = 1$	X_i	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
$S_i = 0$	Y_i	0010	0100	0010	0100	1111	1111	1111	1111	0010	0100	0010	0100	1111	1111	1111	1111
	Z_i	0001	1000	0001	1000	0001	1000	0001	1000	1111	1111	1111	1111	1111	1111	1111	1111

$$A_i^* = \overline{R}_i S_i A_1 + R_i \overline{S}_i A_2 + R_i S_i A_3 + \overline{R}_i A_1 A_2 + \overline{R}_i A_1 A_3 + \overline{S}_i A_2 A_3$$

$$X_i = (\overline{R}_i + S_i)\,(x + A_1 \overline{A}_2 \overline{A}_3 + \overline{A}_1 A_2 A_3)$$

$$Y_i = (\overline{R}_i + \overline{S}_i)\,(y + \overline{A}_1 A_2 \overline{A}_3 + A_1 \overline{A}_2 A_3)$$

$$Z_i = (R_i + \overline{S}_i)\,(z + \overline{A}_1 \overline{A}_2 A_3 + A_1 A_2 \overline{A}_3)$$

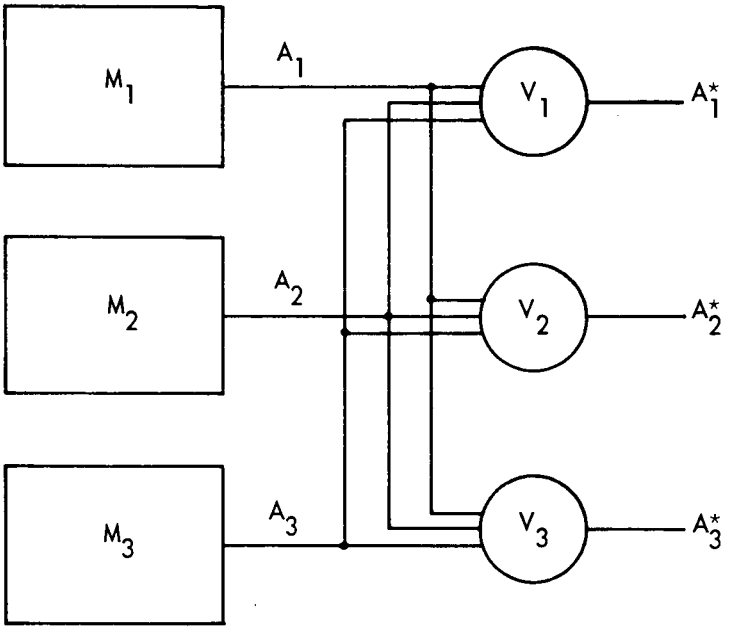


FIGURE 1.–BLOCK DIAGRAM OF TRIPLY
REDUNDANT MODULES WITH TRIPLY REDUNDANT VOTERS

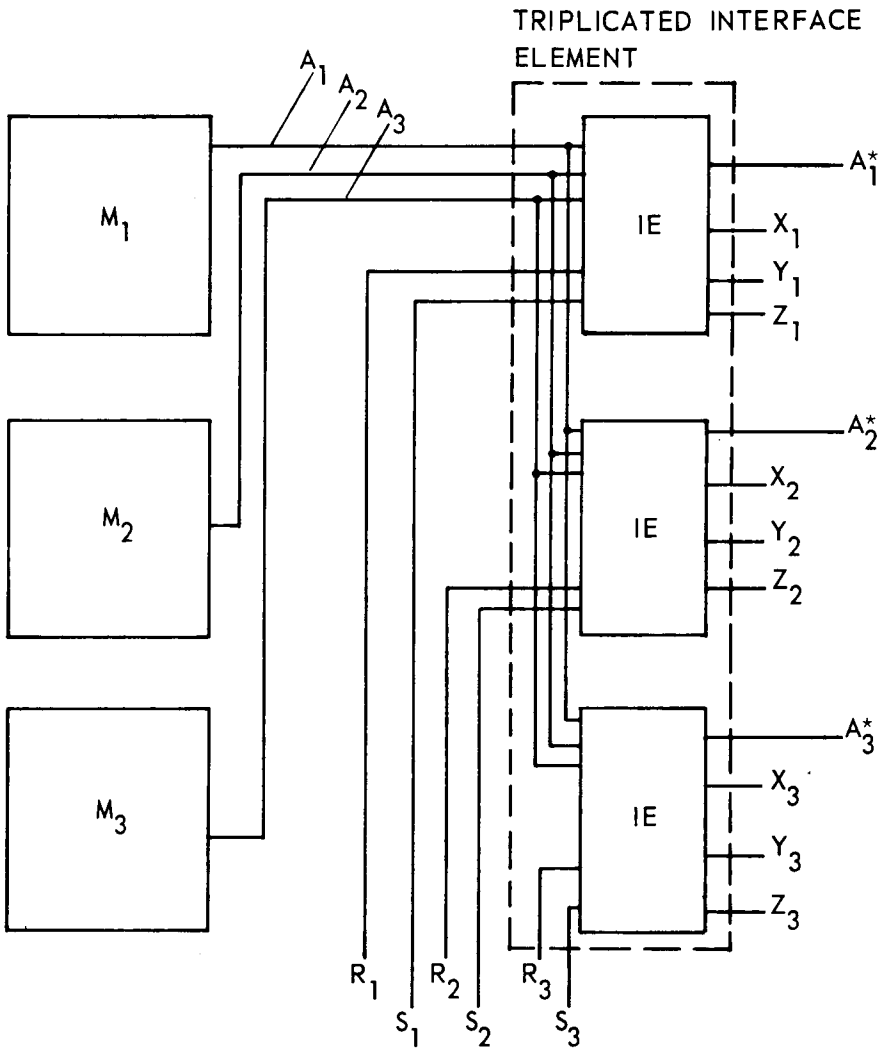
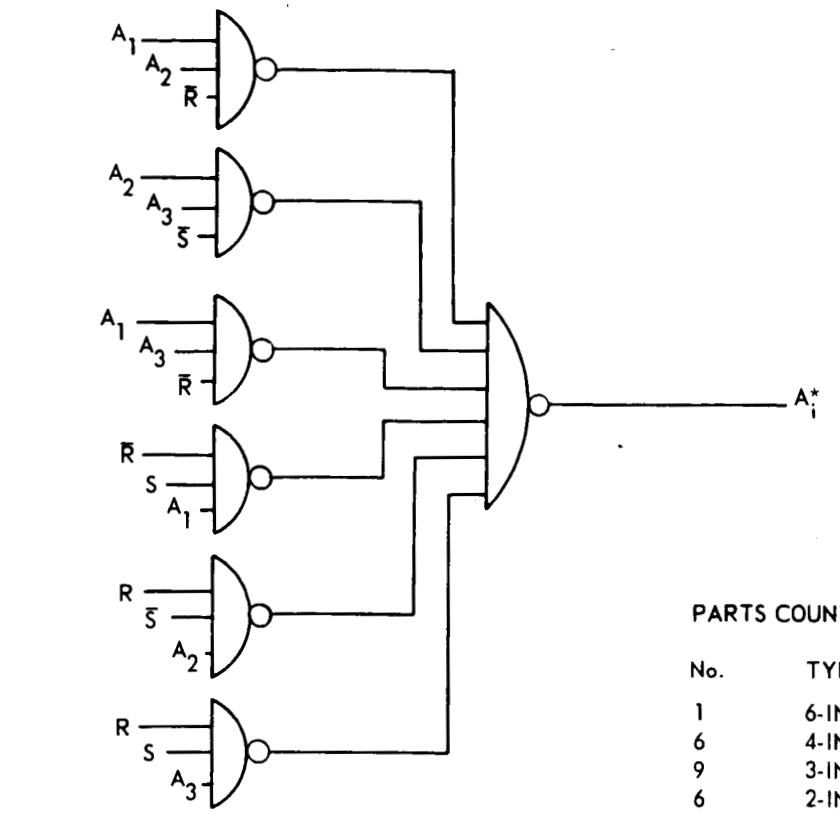


FIGURE 2.–TRIPLICATED MODULES WITH TRIPLICATED INTERFACE ELEMENT

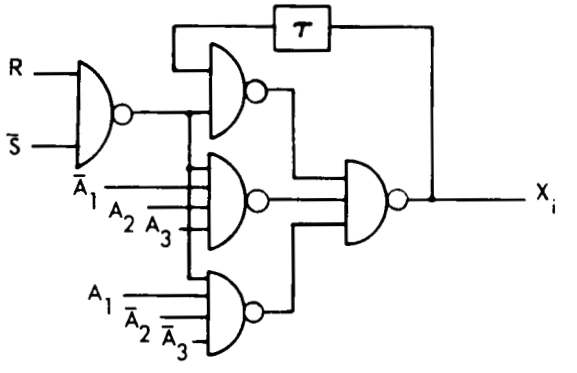


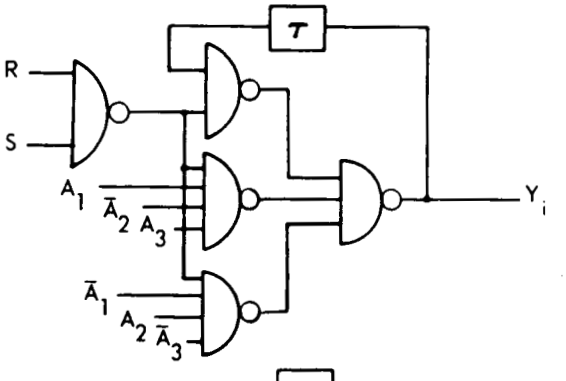
PARTS COUNT

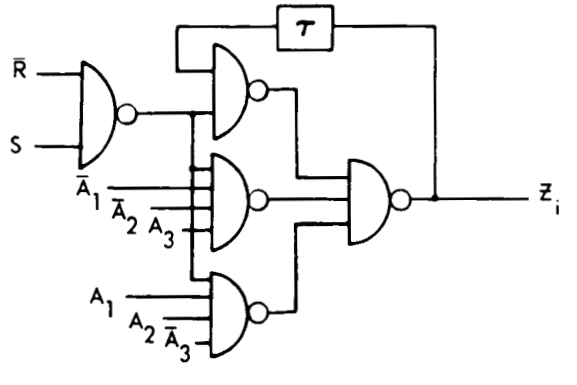
No.	TYPE	INPUTS
1	6-INPUT GATE	6
6	4-INPUT GATES	24
9	3-INPUT GATES	27
6	2-INPUT GATES	12

TOTALS:

22 GATES
69 INPUTS
3 DELAY ELEMENTS
5 INVERTERS
2 LOGIC LEVELS FOR DATA (PLUS ONE FOR CONTROL ONLY)







NOTE: ALL GATES ARE NAND GATES

FIGURE 3–IMPLEMENTATION OF GENERAL IE

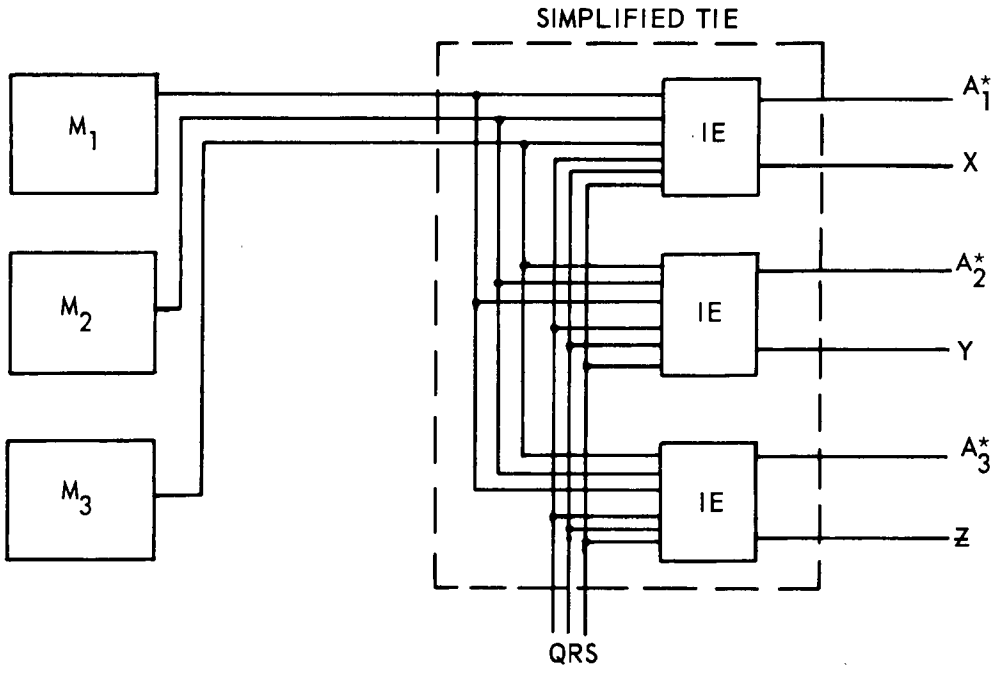
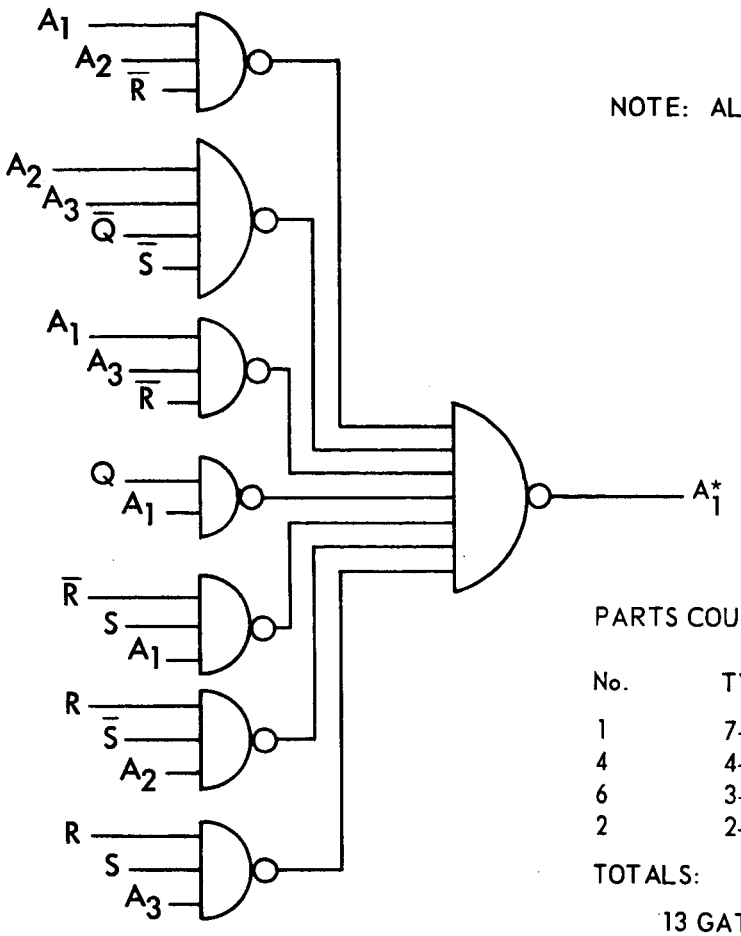


FIGURE 4.–SIMPLIFIED TIE



NOTE: ALL GATES ARE NAND GATES

PARTS COUNT

No.	TYPE	INPUTS
1	7-INPUT GATE	7
4	4-INPUT GATE	16
6	3-INPUT GATE	18
2	2-INPUT GATE	4

TOTALS:

13 GATES
45 INPUTS
1 DELAY ELEMENT
6 INVERTERS
2 LOGIC LEVELS FOR DATA
(PLUS ONE FOR CONTROL ONLY)

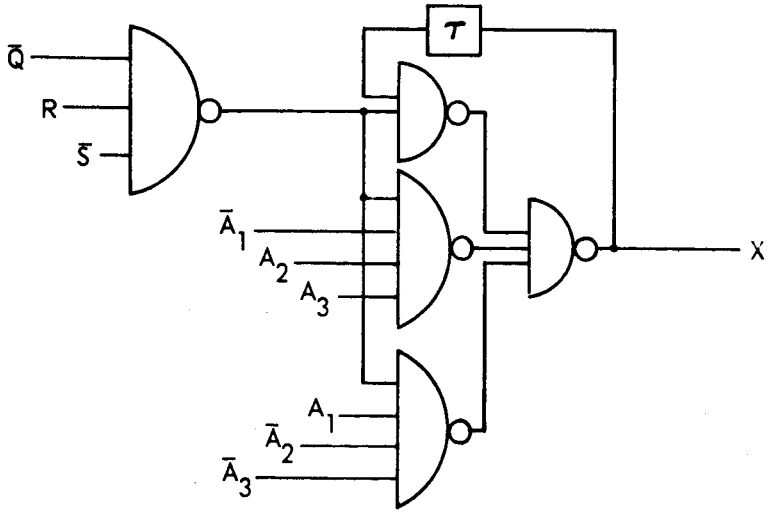


FIGURE 5.–IMPLEMENTATION OF SIMPLIFIED IE FOR OUTPUT A_1^* AND INDICATOR X